Evolution of viral genomes - Interplay between selection, recombination and other forces

Sergei L. Kosakovsky Pond\textsuperscript{1}, Ben Murrell\textsuperscript{2,3}, and Art F. Y. Poon\textsuperscript{4}

\textsuperscript{1} Department of Medicine, University of California, San Diego, USA
\textsuperscript{2} Computer Science Division, Department of Mathematical Sciences, University of Stellenbosch, Stellenbosch, South Africa
\textsuperscript{3} Biomedical Informatics Research Division, eHealth Research and Innovation Platform, Medical Research Council, Tygerberg, South Africa
\textsuperscript{4} BC Centre for Excellence in HIV/AIDS, Vancouver, British Columbia, Canada

1.1 Abstract

RNA viruses evolve very rapidly, often recombine and are subject to strong host (immune response) and anthropogenic (antiretroviral drugs) selective forces. Given their compact and extensively sequenced genomes, comparative analysis of RNA viral data can provide important insights into the molecular mechanisms of adaptation, pathogenicity, immune evasion and drug resistance. In this chapter we present an example-based overview of recent advances in evolutionary models and statistical approaches that enable screening viral alignments for evidence of adaptive change in the presence of recombination, detecting bursts of directional adaptive evolution associated with the phenotypic changes, and detection of co-evolving sites in viral genes.
Keywords: viral evolution, recombination, natural selection, epistasis, machine learning, Bayesian networks.

1.2 Introduction

Whether one considers them to be living organisms or not, viruses are the most extensively sequenced members of the natural world. Virus genomes, especially those of RNA viruses, present many unique challenges to genetic sequence analysis. Even though they are comparably small in size (ranging approximately from $10^3$ to $10^6$ nucleotides in length) and contain a relatively small number of genes, they also undergo a very high mutation rate that drives the accumulation of extensive sequence variation (9). Combined with the extremely rapid pace of evolution due to high mutation and recombination rates, short generation times and strong selection in host environments, viruses provide some of the clearest examples of natural selection in action. Detecting the site-specific signature of selection in viruses by codon-based models of molecular evolution is one of the great achievements of modern evolutionary biology (18, 35). In this chapter, we will cover some of the difficulties often encountered in the analysis of virus genomes and how they may be overcome by recently developed techniques in molecular evolution. Specifically, we will describe and demonstrate methods used to detect recombination, selection, and epistasis from alignments of homologous protein-coding sequences from virus genomes. We will also present a method for identifying factors in the environment (‘agents of selection’) that are responsible for the fitness advantage of certain virus genotypes over others. The reader should be aware that phylogenetics is a rapidly-moving field and that many of the methods being presented in this chapter are relatively new and experimental and consequently have not had time to become well-established in the field. However, we believe that these are the methods that will be of greatest interest to investigators dealing with virus genomic variation.
1.3 Example data and software

Data sets used as examples in this chapter can be downloaded from http://www.hyphy.org/pubs/book2011/data. All computational procedures described below are based on the HyPhy software package (24). A basic level of familiarity with the package is expected and we recommend that readers peruse relevant package documentation, which can be found at http://www.hyphy.org.

1.4 Recombination

We will start by presenting a method for detecting recombination from an alignment of homologous sequences. This is not a conventional ordering of topics because methods for detecting recombination are generally predated by codon model-based methods for detecting diversifying selection (see Section 1.5). However, we strongly advocate screening an alignment for recombination before all else because recombination – which causes different regions of an alignment to be related by different phylogenies – can strongly affect the results of subsequent analyses such as selection detection.

Recombination plays a key role in the evolution of many viral pathogens. For instance, major pandemic strains of the Influenza A virus (IAV) have arisen through segmental reassortment, which can be thought of as intergenic, or gene preserving, recombination. For example, the swine origin H1N1 virus has undergone at least two reassortment events, and carries genes from three different ancestral IAV lineages (53).

In HIV-1 each viral particle packages two RNA genomes and during reverse transcription (RT), the RT enzyme switches between two RNA templates at rates as high as $2 \times 10^{-3}$ per nucleotide per replication cycle (49), creating recombinant DNA templates, which in turn give rise to recombinant progeny. If a single cell is infected with multiple divergent HIV-1 viruses (this can occur in up to 10% of infected hosts (52), depending on a variety of factors), then it is possible that resulting recombinants will found distinct and novel viral lineages. Molecular epidemiology of HIV-1 is replete with examples such lineages, termed Circulating Recombinant Forms (CRFs), with over 40
characterized to date (55).

How frequently recombination occurs is strongly influenced by the viral type and species: (3) found evidence of recombination in 40% of plant RNA genomes that they had examined, but in fewer than 10% of negative sense RNA viruses (4). Apart from its importance in generating novel or removing deleterious genetic diversity and accelerating evolution (58), recombination has a strong effect on many practical aspects of evolutionary analyses (44). As can be seen in Figure 1.1 A, the most apparent effect of including recombinant sequences in a phylogenetic analysis is topological incongruence between trees inferred from different parts of the alignment. In such instances, there is no single topology which can correctly represent the shared ancestry of all the sequences in the sample.

There are many computational approaches to finding evidence of recombination in a sequence alignment (43), however at their core, many such methods look for evidence of phylogenetic incongruence. Here we discuss one such method – GARD, for Genetic Algorithms for Recombination Detection – that we have found to have the best performance among a wide range of approaches on simulated data (26). A genetic algorithm attempts to find an optimal solution to a complex problem by mimicking processes of biological evolution (mutation, recombination and selection) in a population of competing solutions. In this application of genetic algorithms, we are evolving a population of ‘chromosomes’ that specify different numbers and locations of recombination breakpoints in the alignment with the objective of detecting topological incongruence, i.e., support for different phylogenies by separate regions of the alignment. The ‘fitness’ of each chromosome is determined by using maximum likelihood methods to evaluate a separate phylogeny for each non-recombinant fragment defined by the breakpoints (e.g. to the left and to the right of a breakpoint in Figure 1.1), and computing a goodness of fit (small sample Akaike Information Criterion or $AIC_c$) for each such model. The genetic algorithm searches for the number and placement of breakpoints yielding the best $AIC_c$ and also reports confidence values for inferred breakpoint locations based on the contribution of each considered model weighted by how well the model fit the data. For computational expedience, the current implementation of GARD infers topologies for each segment using Neighbor Joining (47) based on the TN93 pairwise distance estimator (54).
1.4. RECOMBINATION

and then fits a user-specified nucleotide evolutionary model using maximum likelihood to obtain AICc scores.

GARD is a computationally intensive method and typically examines $10^3 - 10^5$ competing models on a single dataset. There are two free implementations of GARD, both of which require a distributed computing environment (MPI), in order to fit many models in parallel and speed up the execution: in the HyPhy package (presented here) and on the Datamonkey webserver (http://www.datamonkey.org, discussed in (42)).

We will demonstrate GARD using 13 glycoprotein sequences from Cache Valley Fever virus (CVFv, file CVFg.fas). To execute a GARD screen, launch HyPhy, select Recombination from the standard analyses menu and choose Screen an alignment using GARD / GARD.bf batch file, locate the alignment file and supply values for the following options:

1. **Please enter a 6 character model designation (e.g:010010 defines HKY85)** – this option controls which nucleotide substitution model is to be used for analysis, using PAUP* notational shorthand. The six character shorthand allows the user to specify the entire spectrum from F81 (000000) to GTR (012345), which is a good default option for most analyses. For example, the abbreviation 012232 defines the model with four nucleotide substitution rates: $\theta_{AC}, \theta_{AG}, \theta_{AT}, \theta_{CG} = \theta_{AT}, \theta_{CT}, \theta_{GT} = \theta_{AT}$.

2. **Rate variation options** – determines how site-to-site rate variation should be modeled. Select None to discount site-to-site rate variation; this will cause the analysis to run several times faster than other options, but create the risk of mistaking rate heterogeneity for recombination. This option can only be recommended for alignments with 3 or 4 sequences. Choose General Discrete (the recommended default) to model rate variation using an $N$ bin general discrete distribution, and Beta-Gamma for an adaptively discretized $\Gamma$ distribution (this is a more flexible version of the standard $+\Gamma_4$ model).

3. **How many distribution bins [2-32]** – if rate variation is selected in the previous step, this option allows the user to decide how many different rate classes should be included in the model. We recommend using 3 rate classes by default since both General Discrete and
Beta-Gamma distributions are very flexible and can capture the variability in the majority of alignments with only a few rate classes.

4. **Save results to** – supply a file name where *HyPhy* will write an HTML formatted summary of the analysis. *HyPhy* will generate several other files with names obtained by appending suffixes to the main result file. The `_finalout` file stores the original alignment in NEXUS format with inferred non-recombinant sections of the alignment saved in the ASSUMPTIONS block and trees for inferred for each partition in the TREES block; this file can be input into many recombination-aware analyses in *HyPhy* and other programs that can read NEXUS. The `_ga_details` file contains two lines of information about each model examined by the GA: its $AIC_c$ score and the location of breakpoints in the model. Finally, `_ga_splits` file stores information about the location of breakpoints and trees inferred for each alignment region under the best model found by the GA.

The HTML file generated by a GARD analysis (Figure 1.1 B) presents a summary of the results. In addition to basic model fitting metrics, such as log-likelihood, $AIC_c$, inferred nucleotide substitution rates and site-to-site rate distribution (if selected as an option), the page presents the best-scoring partitioning of the alignment for a given number of breakpoints. For example, among all models with two breakpoints in the Cache Valley Virus glycoprotein alignment, the best model places them at nucleotides 1491 and 1693 and improves the $AIC_c$ over the best model with a single breakpoint (at 1446) by 137.991 points. The score continues to improve until the number of breakpoints reaches 5, at which point the program terminates and reports the best model with 4 breakpoints. If GARD reports that the best model has 0 breakpoints, we may conclude that no evidence of recombination has been found. Note that because genetic algorithms are stochastic, there is no guarantee that replicate runs will converge to exactly the same quantitative results; for example, the difference in $AIC_c$ values between models. When there is a strong signal of recombination breakpoints in the data, however, the qualitative results (number and general location of breakpoints) should be fairly robust.

GARD does not automatically check to ensure that the improvement in model fit is due to a change in the tree topology. For example, if one contiguous part of the alignment evolves at a
1.4. RECOMBINATION

much higher rate than the remainder of the alignment (e.g. an exposed loop), or if the rates of evolution vary among lineages due to heterotachy, then a model which uses two trees with the same topology but different branch lengths may be selected by GARD. To confirm that the topologies differ between segments, it is necessary to execute a post-processing analysis implemented in the Process GARD results / GARDProcessor.bf module. This analysis does not require an MPI environment and must be provided with the same alignment that GARD has been applied to and the .ga_splits file generated by GARD. GARDProcessor.bf performs two tests for topological differences. The first test seeks overall evidence of such differences: it compares the $AIC_c$ score of the best model found by GARD with the fit of the model that uses the same set of breakpoints, but maintains the tree topology inferred from the entire alignment for all partitions. For the CVFg example, the GARD model is strongly preferred by this test. This fact is reported as Versus the single tree/multiple partition model: Delta $AIC = 253.037$. Secondly, the analysis examines whether the trees to the left and right of each breakpoint are topologically different using the Shimodaira-Hasegawa (SH) test (? ), using the RELL approximation scheme (? ) to speed up the calculations. For complete details, please refer to the original GARD manuscript.

In this case, three out of four breakpoints are confirmed using the SH test, with p-values of 0.05 (corrected for multiple testing).

<table>
<thead>
<tr>
<th>Breakpoint</th>
<th>LHS Raw p</th>
<th>LHS adjusted p</th>
<th>RHS Raw p</th>
<th>RHS adjusted p</th>
</tr>
</thead>
<tbody>
<tr>
<td>588</td>
<td>0.00060</td>
<td>0.00480</td>
<td>0.00140</td>
<td>0.01120</td>
</tr>
<tr>
<td>1080</td>
<td>0.00260</td>
<td>0.02080</td>
<td>0.02130</td>
<td>0.17040</td>
</tr>
<tr>
<td>1491</td>
<td>0.00010</td>
<td>0.00080</td>
<td>0.00010</td>
<td>0.00080</td>
</tr>
<tr>
<td>1693</td>
<td>0.00010</td>
<td>0.00080</td>
<td>0.00010</td>
<td>0.00080</td>
</tr>
</tbody>
</table>

To understand the report (p-values will differ slightly between runs because of the stochastic nature of SH resampling) consider the second line: the segment to the left (LHS) of breakpoint 1080 has a topology significantly different from that to the right (Bonferroni corrected p-value of 0.0208), but the reverse is not true (RHS adjusted $p = 0.1704$), hence this breakpoint may be attributed to processes other than recombination.
GARD is geared towards mapping the breakpoints and detecting segments of the alignment which can be adequately described by a single tree topology; as we discuss in the next section, this is necessary to allow more complex analyses to handle alignments with recombinant sequences. Because GARD allows arbitrary tree changes across breakpoints, there are certain cases when it will not perform well; for example, short alignments with many sequences. GARD requires about approximately 4 times as many sites as sequences to run; otherwise the number of samples (sites) is less than the number of model parameters (branch lengths and rates). Another case occurs when only a few sequences in a large alignment have undergone recombination, in which instance the cost of adding many new branch length parameters for one or more trees will likely outweigh the likelihood improvement due to several local subtree rearrangements.

The latter case is common when viral sequences are subtyped. In HIV-1 or IAV, for example, it is common to construct an alignment and a phylogeny of reference sequences with known subtypes or serotypes and then use one of many algorithms to “thread” a sequence to be classified onto the reference topology. One such algorithm is a modification of GARD, called SCUEAL (subtype classification using evolutionary algorithms), developed in (28). Unlike GARD, SCUEAL assumes that the reference sequences can be related by a single topology, which is fixed a priori. It is possible to include recombinant sequences in the reference alignment (see (28) for details). A genetic algorithm searches for the breakpoints in the query sequences only and, for each sequence fragment defined by the breakpoints—the branch in the reference tree where the query sequence attaches. SCUEAL is implemented in HyPhy, and all the necessary files to run it can be downloaded from http://www.hhyphy.org/pubs/SCUEAL/. The download includes a prebuilt reference alignment for HIV-1 pol sequences and documentation on how to make custom reference alignments and screen sequences against them.

1.5 Selection

Selection is the outcome of the variation in fitness induced by the environment in which genetic variants are expressed. Based on the excess number of non-synonymous codon substitutions or
a change in allele frequencies, it is possible to identify sites within protein-coding regions of a genome that have been targeted by selection: some of the methods for accomplishing this are presented in preceding chapters. Diversifying (host-specific) selection on virus genome variation is predominated by the immune response mounted by the host. Jawed vertebrates such as humans have, in addition to the innate immune system, an adaptive immune system that is further partitioned into the humoral and cellular immune responses (31). The humoral response takes place in the extracellular environment and mounts an antibody-based defense that attacks exposed surfaces of the virus particle. The cellular response takes place within the infected cell and involves the recognition and binding of peptides encoded by the virus genome, which are displayed on the surface of the cell to trigger the lysis of the cell by cytotoxic T-lymphocytes (CTLs). Both components of the adaptive immune system play a crucial role in managing a viral infection and thereby shaping the genetic variation of the virus population. In addition, many human pathogenic viruses, particularly HIV-1, influenza virus, hepacivirus and herpesvirus, are treated by antiviral agents that also target specific sites of the virus genome (17).

1.6 Detecting selection in the presence of recombination

In order to infer selection in an alignment or at individual sites, most algorithms estimate the rates of synonymous and non-synonymous substitutions and test them for equality. It has long been recognized that by confounding the phylogenetic signal, recombination can mislead rate estimation procedures and natural selection tests, often severely (1). A simple illustration of this can effect be seen in Figure 1.2.

The simple approach to guard against this undesirable behavior is to identify and remove recombinant sequences prior to running selection analyses. In addition to practical difficulties in reliably detecting which sequences have been subject to recombination, discarding sequence data lowers power of analyses and could introduce unanticipated biases. (48) proposed a PARtitioning approach for Robust Inference of Selection (PARRIS) to retain all the sequences, including recombinants, for selection testing, whose stages are described below.
1. The input alignment is screened for evidence of recombination, e.g. using GARD, and the number and location of breakpoints are inferred.

2. A separate tree is constructed for each non-recombinant segment; for the CFVg alignment, this would generate five alignment segments and five corresponding trees.

3. A codon model (see previous sections or (7) for further details) is defined using the following rate matrix, whose $q_{ij}$ element describes the instantaneous rate of substitution between of codons $i$ to codon $j$:

$$q_{ij} = \begin{cases} 
\alpha \theta_{ij} \pi_{ij}, & \text{a single nucleotide synonymous substitution,} \\
\omega \alpha \theta_{ij} \pi_{ij}, & \text{a single nucleotide non-synonymous substitution,} \\
0, & \text{multiple nucleotide synonymous substitutions required,} \\
-\sum_{k \neq i} q_{ik}, & i = j.
\end{cases}$$

$\theta_{ij}$ parameterize the unequal substitution rates between nucleotides, $\pi_{ij}$ are the frequency parameters, correcting for the nuleotide composition of the alignment, $\alpha$ – the synonymous substitution rate and $\omega$ – the familiar ratio of non-synonymous to synonymous substitution rates. For example, $q_{AC,A,TA} = \alpha \omega \theta_{CT} \pi_{CT}^2, q_{AC,T,ACA} = \alpha \theta_{AT} \pi_A^3, q_{AAA,CCC} = 0$. Notice that $\theta_{ij} = \theta_{ji}$ (because of time-reversibility of the process); $\pi_{n}^m$ refers to the observed frequency of nucleotide $n$ in codon position $m$ (the MG frequency parametrization (7)). One key feature of this model is that both $\alpha$ and $\omega$ can vary from site to site; traditionally it has been assumed that $\alpha$ is proportional to the mutation rate and is constant across all sites. There is increasing evidence that synonymous rates vary among sites as well, e.g due to secondary structure of viral RNA, codon usage bias, and not accounting for such variation can cause misidentification of relaxed constrains as positive selection in some cases (e.g. see (38)).

4. All parameters of the codon model are estimated jointly from all non-recombinant data partitions, while the tree topology and branch lengths are allowed to differ between partitions. In this way, recombination is accommodated (different topologies and branch lengths), but the parameters of the evolutionary process (e.g. $\omega$) are inferred from all sequences jointly.
1.6. DETECTING SELECTION IN THE PRESENCE OF RECOMBINATION

5. Two models with site-to-site rate variation are fitted to the data: the null model which restricts \( \omega \leq 1 \) and the alternative model which does not have this restriction. The models are analogous to M1a and M2a implemented in the PAML package (described in an earlier section), except that in PARRIS, synonymous rates \( \alpha \) are also variable, and drawn from a 3-bin general discrete distribution.

To start a PARRIS analysis, launch HyPhy, select Selection/Recombination from the standard analyses menu and choose A PARTitioning approach for Robust Inference of Selection / PARRIS.bf batch file, supply an alignment file and choose values for one of the many analysis options. In addition to the models of (48), PARRIS.bf implements those described in (38), adapted for handling partitioned data, and a number of unpublished or experimental options.

1. **Choose Genetic Code** – select the genetic code appropriate for the alignment under investigation.

2. **How many datafiles are to be analyzed?** – PARRIS and other recombination-aware selection analyses (see Exercises) can read NEXUS formatted files with multiple partitions encoded in the ASSUMPTIONS block and corresponding trees in the TREES block (e.g. the _finalout files output by GARD), or read individual partition and tree files. For the latter option, select the number (\( \geq 1 \)) of files to be input, and for the former – enter 1.

3. **Branch Lengths** – to speed up calculations, HyPhy can use branch lengths estimated from a nucleotide model for the analysis, i.e. hold them constant while the codon model is being fitted (the Nucleotide Model option, suitable for initial screens, especially on larger alignments), or estimate them together with all other parameters (the Codon Model option, suggested to confirm results).

4. **Options for handling equilibrium frequencies** – select how to parameterize codon frequencies in the substitution model, Muse-Gaut (MG) vs Goldman-Yang (GY). There are some reason to prefer MG in general (see (7) for a discussion).
5. **Nucleotide Rate Matrix Options** – specify the nucleotide bias component of the substitution model.

6. **Options for multiple classes of non-synonymous substitutions** – decide how the model will handle unequal substitution rates between different amino-acids. **Single** specifies that a single $\omega$ rate will apply to all non-synonymous substitutions (this is by far the most common option). With **Multi**, the analysis will prompt the user to select a file defining the protein analog of the (012232) string for nucleotide models as a $20 \times 20$ matrix (see the Mutlirate.mdl file for an example). There are a number of ways such a matrix can be obtained, including a model selection process for codon data (8). **NMulti** allows the specification of numerical substitution rates between pairs of amino-acids (much like the BLOSUM62 matrix used by blastp).

7. **Rate Variation Models** – allows rate variation models described in (38) to use the partitioning approach. For PARRIS analyses, select the **Dual** option, where both synonymous and non-synonymous rates vary from site to site.

8. **Independent or multiplicative nonsynonymous rate** – in the rate matrix defined above, we parameterized the non-synonymous substitution rate as $\alpha \omega$, i.e. via a multiplicative factor which modulates the synonymous rate (the **Multiplicative** option). It is also possible to parameterize this rate via an independent parameter $\beta$ (the **Independent** option). The latter is generally more flexible, e.g. it allows both $\alpha = 0$ and $\beta > 0$ which cannot be parameterized through a finite $\omega$ ratio, but makes the testing for selection (i.e. $\omega > 1$) difficult. The PARRIS analysis uses the **Multiplicative** option.

9. **Codon or nucleotide level synonymous rate variation** – this is an experimental (at the time of writing) option. Select **Codon (syn1)** to run PARRIS. The other option **Nucleotide (syn3)** allows the model to vary synonymous substitution rates ($\alpha$) based on the position of the codon where the substitution is taking place.

10. **Distribution Options** – determines which site to site rate variation models will be fitted to
the data. The PARRIS option runs the two discrete models needed to test for evidence of diversifying positive selection described in the original manuscript, while the others provide more choices, including discretized gamma distributions. Selecting Run All or Run Custom provides access to all or some of these models.

11. **Initial Value Options** – allows the optimization procedure to start from predefined values (Default) or from a random starting point (Randomized). The latter option is useful for checking convergence; if multiple runs of the analysis attain the same log likelihood and parameter values, then the procedure has converged.

12. **Save summary result file** – HyPhy will write analysis summary (also echoed to the screen) to this file. Also, it will create three files for each of the fitted model, by appending suffixes to the summary file name, much like in GARD. For PARRIS, the null model is named M1a and the alternative model – M2a. The .model.fit file contains the fitted likelihood function for each model in a NEXUS format with the HYPHY block used to encode the model and parameter estimates. The .model.distributions file stores a text summary of the distributions of synonymous and non-synonymous rates inferred for the model, while the .model.marginals file provides a detailed report for the empirical Bayes analysis carried out by the program to identify sites subject to negative and positive selection and posterior distributions of $\omega$ and $\alpha$ values at each site.

As an illustration, we will run the PARRIS analysis with the REV nucleotide model, Codon branch lengths, MG frequency option, single non-synonymous rate class, dual rate variation model, multiplicative nonsynonymous rate, codon level synonymous rates, PARRIS distributions and default starting values on HepatitisE.nex (single partition) and HepatitisEgard.nex (GARD-inferred .finalout) files, containing an alignment of 21 capsid sequences from Hepatitis E virus. PARRIS executed on the unpartitioned alignment provides the following summary output:

```
<table>
<thead>
<tr>
<th>Model</th>
<th>Log Likelihood</th>
<th>Synonymous CV</th>
<th>NS Exp and CV</th>
<th>N/S Exp and CV</th>
<th>P-Value</th>
<th>Prm</th>
<th>AIC</th>
</tr>
</thead>
</table>
```
In this particular instance, allowing a proportion of sites to evolve with $\omega > 1$ (M2a model) provides a significantly improved fit compared to the null model which only permits sites with $\omega \leq 1$, both according to the likelihood ratio test ($p = 0.007$ based on the $\chi^2$ distribution) and Akaike Information Criterion (21331.69 vs 21337.71). The other values reported in the table summarize means and coefficients of variation (CV) for synonymous and non-synonymous distributions of rates. In .M2a.marginals file, four sites are reported to be under diversifying positive selection with posterior probabilities of 0.95 or greater (23,109,110 and 115).

GARD inferred 4 breakpoints (5 partitions) in this data set, and the corresponding summary table is

<table>
<thead>
<tr>
<th>Model</th>
<th>Log Likelihood</th>
<th>Synonymous CV</th>
<th>NS Exp and CV</th>
<th>N/S Exp and CV</th>
<th>P-Value</th>
<th>Prm</th>
<th>AIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>discr(3), M1a</td>
<td>-10457.49607</td>
<td>0.42057602</td>
<td>0.08316,3.16349</td>
<td>0.20055, 7.23691</td>
<td>N/A</td>
<td>84</td>
<td>21082.99</td>
</tr>
<tr>
<td>discr(3), M2a</td>
<td>-10454.94654</td>
<td>0.42703723</td>
<td>0.11230,3.68455</td>
<td>0.28263, 8.72351</td>
<td>0.0781191</td>
<td>86</td>
<td>21081.89</td>
</tr>
</tbody>
</table>

Notice that the evidence for positive selection is much weaker when recombination is taken into account: LRT p-value is no longer significant and the AIC improvement is much smaller, compared to the unpartitioned analysis. The partitioned models has a much better AIC than their unpartitioned counterparts, indicating that the data are better explained by the former. Also, no positively selected sites with posterior probabilities of 0.95 or greater are found.

This example demonstrates that if recombination could have shaped the evolutionary history of sequences being analyzed, it is prudent to use approaches which take it into consideration, lest it be misinterpreted as another process, e.g. positive selection. All selection analyses in HyPhy and Datamonkey accept partitioned data, thus allowing researchers to keep all the sequences and correct for the confounding effects (see Exercises for another example).
1.7 Directional selection

HIV-1 replicates extremely rapidly, producing as many as $10^{10}$ viral particles per day. The fidelity of reverse transcription is low, with a rate of $3 \times 10^{-5}$ errors per base per replication cycle. Together, this provides HIV-1 with a powerful means to escape selective pressure introduced by antiretroviral therapy (ART), which suppresses HIV-1 replication by interfering with various stages of the viral life cycle, leading to drug resistance.

Some important features of the evolution of drug resistance must be encoded by models of evolution to detect substitutions under selective pressure induced by ART. For this discussion, we are modeling evolution over a reverse transcriptase phylogeny that has been constructed from treatment naive, as well as post-treatment sequences (see figure 1.3). The first thing to notice is that the selective pressure of interest is not constant over the entire phylogeny, but rather restricted to a subset of branches: it is episodic. A second critical property of the evolution of drug resistance is that once ART is introduced, selection is directional, where only substitutions towards one or more target amino acids are favored. This can be contrasted with diversifying selection, where nucleotide substitutions that change the amino acid are favored, regardless of the amino acid. Diversifying selection approximates the continuously shifting co-evolutionary environment typified by host-pathogen “arms-race” co-evolution (21). The evolution of drug resistance, on the other hand, is characterized by discrete major shifts of fitness landscape with the introduction of therapies. The probability of the emergence of particular amino acids contributing to drug resistance increases inexorably with time, as long as viral replication is not suppressed. Once treatment resistance emerges, selection becomes purifying as long as the drug regimen is maintained.

MEDS (a Model of Episodic Directional Selection) models directional selection along a priori selected foreground branches, while assuming the background branches evolve in a non-directional (but not necessarily neutral) manner. MEDS is a codon model, based on $MG94 \times REV$ (which combines a general time-reversible model of nucleotide substitution with separate synonymous and non-synonymous rates, $\alpha$ and $\beta$), that extends two earlier models of coding sequence evolution: 1) the episodic component of MEDS is structurally identical to the Internal Fixed Effects
Likelihood (IFEL) model proposed by (25), and, 2) the directional component is introduced in the same manner as the model of directional selection proposed by (50).

Two separate codon models are used to model substitutions along foreground and background branches. A single synonymous rate $\alpha$ is shared between them, but each is allowed its own non-synonymous substitution rates ($\beta^F$ and $\beta^B$). Diversifying selection is thus allowed on both foreground and background branches. Directional selection along foreground branches is introduced with $\omega_T$, which is multiplied onto the rates of all substitution towards to a specified target amino acid $T$. Elevating $\omega_T$ will thus increase the rate of substitutions to $T$. The analysis proceeds site-by-site. Branch lengths and nucleotide rate parameters are first estimated from the whole alignment under a simpler model. For each site, we define the null model by setting $\omega_T = 1$, a special case of the alternative model where $\omega_T$ is free to vary. The null model has 3 free parameters per site: $\alpha$, $\beta^F$ and $\beta^B$. The alternative model has a single additional parameter, $\omega_T$, biasing substitutions towards $T$. To test for selection towards amino acid $T$ at a specific site, we obtain maximum likelihood scores for the null and alternative models and perform a likelihood-ratio test (LRT). Scanning a site for selection towards any possible amino acid ($T$) involves testing 20 hypotheses, and Bonferroni correction (46) is employed to control the site-wise Type I error rate.

To run a MEDS analysis, an alignment and rooted phylogeny are required. Furthermore, the foreground branches of the phylogeny must be labeled. To do this, $\{\text{FG}\}$ is placed after the foreground node names (but before the colons) in the Newick tree string. In this example (sub)tree, Branch1 is labeled foreground: (Branch1\{FG\}:0.1,Branch2:0.1); For large trees, editing the Newick files by hand is inefficient. One solution is to use FigTree (http://tree.bio.ed.ac.uk/software/figtree/) to color the foreground branches, and then replace the color tag (e.g: [\&!color#-64512]) in the resulting Newick string with $\{\text{FG}\}$. Once the phylogeny is suitably annotated, from Hyphy execute Standard Analyses / Positive Selection / MEDS.bf, select the data and tree files, and specify an output .csv file. The output file contains the maximum likelihood parameter values and LRTs for each of 20 amino acids at each site. To assist with interpreting such a large file, we provide a web script (www.cs.sun.ac.za/~bmurrell/py/MEDSproc) that takes an output file and p-value threshold and summarizes all
detected substitutions. In addition to the test for directional selection, MEDS.bf also performs a test for episodic diversifying selection along foreground branches. These results are included in the output and summary files.

Table 1.1 displays the results for MEDS on the reverse transcriptase alignment (HIV_RT.fasta) for the phylogeny in figure 1.3 (HIV_RT_tagged.tre). This alignment contains 26 sequences from patients before the initiation of ART, and after failing ART, obtained from the Stanford HIV Drug Resistance Database (hivdb.stanford.edu). We tested for episodic directional selection with MEDS, episodic diversifying selection (which MEDS.bf automatically tests for), and constant diversifying selection, using FEL from Datamonkey. Using a p-value threshold of 0.05, MEDS detected 7 substitutions under selection, 6 of which are known drug resistance associated mutations (DRAMs). The test for episodic diversifying selection detected 5 sites under selection, all known to be associated with drug resistance. The test for constant diversifying selection detected only 2 sites, both involved in drug resistance. On this alignment, the performance of MEDS and its accompanying test for episodic diversifying selection were similar, and both clearly outperformed the FEL test for constant diversifying selection. On other datasets, greater differences between MEDS and the test for episodic diversifying selection have been observed. In one case (32), on a much larger RT phylogeny, MEDS detected 16 substitutions (13 DRAMs) while the test for episodic diversifying selection identified only 4 sites (2 DRAMs). The factors contributing to the performance differences between datasets are still being explored.

Another model, EDEPS (Episodic Directional Evolution of Protein Sequences), was proposed in (32). EDEPS extends DEPS, proposed by (27). EDEPS also detects sites with increased substitution rates towards specific amino acids, but it differs from MEDS in two ways: 1) EDEPS models directional selection of amino acid rather than codon sequences, 2) EDEPS uses a Random Effects Likelihood (REL) framework to bias selection towards amino acids across all sites, relying on an empirical Bayes analysis to identify sites of interest. As in MEDS, accelerated substitutions towards a target residue $T$ are restricted to foreground branches. Background branches evolve according to a baseline protein substitution model, which, for this task, would be the HIV-Between empirical model (33). It is well known that amino acid substitution rates depend on the residues
involved (e.g. see (8)), and specifying a baseline model which includes unequal substitution rates provides a qualitative advance over MEDS. Conversely, because EDEPS works with protein sequences, the natural proxy of neutral evolution is not available. The performance of EDEPS is similar to MEDS on most datasets tested so far.

MEDS and EDEPS are applicable whenever a set of a priori known branches on a phylogenetic tree are expected to be under the same or similar kinds of selective pressure. The power to detect directional selection on foreground branches is likely to decrease with their quantity, although this has not been tested explicitly. Also important is the arrangement of the foreground branches: if the difference between amino acids on foreground and background branches can be explained by a single substitution along a single branch, then there will be little evidence for directional selection. Paired HIV sequences sampled from the same patient before and after therapy produce an ideal arrangement of disconnected foreground branches, although MEDS and EDEPS still perform well in less ideal cases.

1.8 Epistasis

The effect of a mutation depends not only on the host environment, but also on the rest of the genome sequence in which it occurs. Put another way, the rest of the genome comprises an extremely significant part of the mutation’s environment. The dependence of a mutation’s effect on other sites of the genome is known as epistasis. Because epistasis is inherently non-linear, it is exceedingly difficult to model and hence to estimate from data. In quantitative genetics, epistasis is assessed as a non-additive component of variance attributable to interactions among genetic factors (15); however, this framework does not provide a means of explicitly identifying those interactions. On the other extreme, population genetics models tend to incorporate epistasis as a non-additive term for the effects of mutant alleles at two loci (6). While this scheme is mathematically convenient, it is not adequate for the purpose of studying the evolution of genomes, even when they are relatively small in size.

The comparative study of sequence variation offers a practical approach to identifying which
sites in the genome participate in epistatic interactions. Literally hundreds of investigators across disjoint sub-disciplines of biology have proposed various comparative methods to accomplish this objective. Though we have not yet encountered a comprehensive review, interested readers may find useful references in (2, 5, 19, 40). Essentially all of these methods use correlated patterns of substitution at different sites as evidence of an interaction. Most methods apply some correlation test statistic to pairs of amino acid sites (columns) in an alignment of protein sequences. (Göbel and colleagues (14) are often cited as the first example of this approach, but they were in fact preceded by Korber and colleagues (22).) While this is a convenient approach, it ignores the biological reality that sequences are the product of evolution. In other words, a significant correlation between sites can easily be confounded by the evolutionary relationships among the observed sequences, especially when certain combinations of residues have been inherited by large numbers of descendants without further modification (11). A statistical association between codon sites in a gene would then be falsely attributed to some functional interaction between residues (‘identity by state’), when it was in fact due to evolutionary relationships (‘identity by descent’).

To overcome the confounding effect of evolutionary history, we have advocated and extended Felsenstein’s approach of redirecting the focus of comparative study from patterns in the end products of evolution, to patterns in the process of evolution itself (40, 41). We are looking for residues that ‘co-evolve’, such that a substitution at one site accelerates the substitution rate at one or more other sites. Let’s proceed with an analysis of the example file p24.seq, which contains an alignment of HIV-1 subtype C gag p24 sequences and a tree. Because this analysis will use many of the same functions as many of the standard selection analyses implemented in HyPhy, a template for detecting co-evolution (epistasis) can be accessed though the QuickSelectionDetection.bf batch file listed in the Standard Analyses menu under the heading ‘Positive Selection’. In summary, a codon substitution model will be fit to the sequence alignment and a tree. Maximum likelihood parameter estimates are subsequently used to reconstruct ancestral sequences at the internal nodes of the tree using a fast algorithm proposed by (45). It is then straight-forward to map mutations to branches of the tree by comparing character states at each codon site at the start and end of each branch. If the reconstructed/observed character states are different, then a mutation must have
occurred at some point along the branch (34, 56). This mapping procedure does not account for
cases where more than one substitution occurs at the same codon site in a branch, but see (10) for
a method that can account for these ‘multiple hits’. When the distributions of mutations mapped
to branches of the tree are significantly correlated between two codon sites, then we interpret this
outcome as evidence of an epistatic interaction between those sites.

We could then proceed by comparing mutational maps between all pairs of codon sites in the
alignment (20, 51). While this is a convenient framework for identifying sites with interactions,
however, it is subject to the problem of ‘confounding’. To use a popular example in the artificial
intelligence literature, a pair-wise analysis will find a significant association between sales of ice-
cream and the number of drownings at a public pool. A naïve observer would then be led to believe
that consuming ice-cream causes one to drown. Of course, what is actually happening is that ice-
cream sales and the number of swimmers at the pool are both greater on warm and sunny days,
and that drowning is more frequent when more people are swimming. It is difficult to reconstruct
this system of cause-and-effect by taking an approach that evaluates only pairs of variables at a
time. Following through with our analogy, an agent of selection is like our ‘sunny days’ variable
that is the common cause of multiple effects (different sites in the genome, like ‘ice-cream sales’
and ‘swimming’ variables). If we are limited to evaluating pairs of variables at a time, we may
consequently be led to falsely report an epistatic interaction between sites in the genome. Similarly,
we could over-estimate the number of sites influenced by an agent of selection because of actual
epistatic interactions among those sites.

To account for this problem of confounding, we use Bayesian networks to analyze the joint dis-
tributions of mutations that we have mapped to the tree at all sites by maximum likelihood-based.
A Bayesian network is a graphical model giving a compact representation of the joint distribution
of variables (37). It is comprised of ‘nodes’ that represent variables, and ‘edges’ that connect
nodes to indicate that one variable is conditionally dependent on another. In our context, the nodes
correspond to mutational maps for different columns (codon sites) in the alignment or the pres-
ence/absence of a selective agent, and edges correspond to a statistical association between maps.
By evaluating the joint distribution, we are assessing all the variables at once and are therefore able
to discriminate between real associations (sunny days and ice-cream) and those that are spurious (ice-cream and drowning). Our ideal objective is to find the Bayesian network that best explains the data. However, the space of all possible Bayesian networks is so astronomically large for a modest number of variables that we must abandon any hope of finding a single best network. Instead, we take a Bayesian approach and endeavour to generate a random sample of Bayesian networks from the posterior distribution that is shaped by the data (12). (In fact, we follow (12) and further collapse the space of Bayesian networks into a new space over permutations of ‘node orders’, i.e., assertions about which other nodes in the network a given node can be conditionally dependent on. This transformation greatly reduces the size of model space and smooths the posterior probability surface as well, resulting in better model convergence.) A random sample is obtained by Markov chain Monte Carlo (MCMC) using the Metropolis-Hastings sampling algorithm (16, 30), which essentially explores the posterior probability surface by taking a biased random walk from one point in model space to another.

Execute the **QuickSelectionDetection.bf** batch file to analyze the **p24.seq** file. This first set of options are the same for analyses of positive selection, and may be familiar to you:

1. **Choose Genetic Code** - Because HIV replicates in human hosts, choose the ‘Universal’ option.

2. **New/Restore** - This menu gives you the opportunity to reload a nucleotide model fit from a previous execution of QuickSelectionDetection.bf on the same alignment and tree, which can save computational time. Since this is probably the first time you have run this batch file on these data, select ’New’.

3. Select the **p24.seq** file.

4. **Model Options** - Specification of the nucleotide and codon substitution models. The ‘Default’ is to fit the HKY85 nucleotide substitution model to refine estimates of branch lengths in the tree, followed by fitting the Muse-Gaut codon substitution model crossed with the HKY85 model with branch lengths in the tree constrained to scale by a factor that is estimated from the data.
5. Enter ‘y’ in the console window to use the tree included with the FASTA file.

6. Specify a file name to export the maximum likelihood fit of the nucleotide model.

7. **dN/dS bias parameter options** - Select ‘Estimate dN/dS only’.

   At this point, the batch file branches into different types of analyses with very different options. You should select the option ‘BGM co-evolution’ (Bayesian Graphical Model (BGM) is a synonym for a Bayesian network) in order to perform ancestral reconstruction and mutational mapping followed by Bayesian network analysis. These are the analysis options that are raised by the BGM co-evolution pipeline:

   1. **Treatment of Ambiguities** - Ambiguous nucleotide calls can be interpreted according to one of two extreme assumptions — either that they are all errors and should be resolved to the predominant nucleotide in the alignment (Resolved), or that they reflect genuine polymorphisms in the population (Averaged). This assumption will affect how ancestral sequences are reconstructed at internal nodes of the tree.

   2. **Substitution count cutoff** – The HyPhy console window will display some descriptive statistics of the distribution in the number of branches that substitutions have been mapped for all sites in the alignment. For our p24 example, the mean and medians of this distribution are 12.3 and 2, respectively. This indicates that most codon sites have two or fewer substitutions mapped to the tree, while there are a small number of highly variable sites inflating the mean. This is useful information because it indicates that we can ignore the majority of codon sites in the alignment, either because they are completely conserved or because there is insufficient variation to detect co-evolution (i.e., one or two substitutions). As a very rough rule-of-thumb, we don’t like to have more variables sent to the Bayesian network than we have observations. The p24 alignment is comprised of 541 sequences; a cut-off of 10 mapped substitutions per site leaves 51 codon sites to analyze.

   3. **Maximum parents** — We assume that nodes cannot be conditionally dependent on more than $X$ other nodes (parents), where $X$ is either 1 or 2 in this template batch file. For
example, the space of all possible networks includes those in which one variable is asserted to depend on every other variable. Such cases are either unlikely or not very informative, and this simplifying assumption drastically reduces the space of all possible Bayesian networks. For a quick preliminary analysis, you should select a limit of 1.

4. **Duration of MCMC chain** — This needs to be a large number in order for the Metropolis-Hastings sampler to explore the parameter space long enough to obtain an adequate sample, given that auto-correlation in the chain sample is inevitable, *i.e.*, that adjacent states in the chain will be very similar. The default value of $10^5$ steps is adequate for most 1-parent networks. A 2-parent network analysis will require a longer chain and burn-in (see next item).

5. **Duration of MCMC burn-in** — The number of steps that will be discarded as a burn-in period, *i.e.*, for the sampler to move into favorable regions of model space with a relatively high posterior density. Note that this number must be smaller than the previous setting (you cannot discard more steps than you have run the chain for.)

6. **Sampling interval** — The length of the interval for ‘thinning’ the chain. Because the chain is inevitably highly auto-correlated, it is necessary to thin the chain sample by taking every $n$-th step. The default interval size of 1000 steps is reasonable and results in a final sample size of 90 under all default settings, *i.e.*, $(10^5 - 10^4)/1000$.

7. **Ancestral resampling** — The reconstruction of ancestral sequences by maximum likelihood is increasingly uncertain as we go deeper into the tree, *i.e.*, further from the observed sequences. To address this uncertainty, we provide an option of using a non-parametric bootstrap procedure to re-sample ancestral sequences from the posterior probability distributions of character states at each internal node of the tree. The Bayesian network analysis can then be run on each of these bootstrap samples. Because the computational time scales linearly with the number of samples, we provide a message-passing interface (MPI) implementation that can distribute MCMC runs across processors; however, this requires that you are running an MPI-enabled command-line build of HyPhy, *i.e.*, HYPHYMPI.
8. **Output files** — You will be prompted to identify two files to write analytical results to. The first file will contain the edge marginal posterior probabilities, \textit{i.e.}, the proportion of the chain sample that contains a given edge. The second file will contain a graph (encoded in the DOT language interpreted by GraphViz, which is open-source software for rendering graphs that can be downloaded from http://www.graphviz.org) comprised of all edges with marginal posterior probabilities exceeding a cutoff of 95%.

When you have run through this analysis using the \textit{gag} p24 example data, you will find that \textit{HyPhy} has spawned two new windows. One window is labeled ‘MCMC Trace’ and displays the chain sample that has been thinned down to 90 steps (Figure 1.4). By default, the chain is displayed as a scatterplot, but it is easy to switch to a step plot by selecting this option from the drop-down menu labeled ‘Type’. This plot is a convenient means for spotting severe cases of autocorrelation, \textit{i.e.}, a clear trend of increase and/or decrease over the length of the thinned sample. For example, if there was a clear monotonic increase in the sample over time, then it would be highly likely that the sample size was inadequate and the model would need to be run for substantially longer duration.

A second window displays a histogram summarizing the edge marginal posterior probabilities in the thinned sample (Figure 1.4). A U-shaped distribution indicates that there is sufficient data to identify a minority of edges that are highly likely to be in the Bayesian network from a background of edges that are highly unlikely to occur. This distribution can be used to customize the cutoff value (default 0.95) used to generate a consensus Bayesian network (Figure 1.5). For example, the histogram in Figure 1.4 suggests that there are many edges with marginal posterior probabilities between 0.85 and 0.95 that one might be interested in seeing included into the default network.

### 1.9 Identifying agents of selection - the CTL response

In previous sections, we have outlined several methods for detecting the signature of selection from an alignment of homologous sequences. It is much more difficult to identify which aspects of the host environment were responsible for favoring one variant of a genome over another. These external factors are generally known as the “agents of selection” (57). For example, the cellular
immune response identifies infected host cells for lysis by cytotoxic T-lymphocytes (CTLs), based on the presence of peptides derived from virus proteins by the antigen processing pathway in the cell. In human cells, 9-mer peptides are recognized and bound by human leukocyte antigen (HLA) molecules that are encoded by the highly-variable major histocompatibility class (MHC) I loci. Consequently, many sites in a virus genome experience strong selection for amino acid replacements because they encode components of a protein that are preferentially targeted by the antigen processing pathway, such as the ‘anchor’ residues that determine HLA binding specificities. We would like to know which regions of a virus genome are enriched for sites targeted by the cellular immune response — such regions can identify peptides to be incorporated into anti-HIV vaccine candidates (13). However, there are hundreds of alleles that have been described at the three MHC class I loci (denoted A, B and C) and each one can potentially target a different set of sites in the HIV-1 genome.

This is a situation that is amenable to being analyzed with a Bayesian network because potential agents of selection in the host environment can simply be handled as additional variables in the graph (39). Simply put, we want to know if substitutions tend to occur more often than random in branches that represent hosts that are presenting a particular agent of selection. The capacity of Bayesian networks to find causal relationships in the midst of potential confounding variables is an important strength of this application. However, there is a catch — we cannot reconstruct host environments in the virus phylogeny. This limits an analysis of associations between agents of selection and site-specific rates of virus evolution to the terminal branches of the tree; in other words, the branches that are leading directly to observed virus sequences. Unfortunately, that means that we must sacrifice a substantial amount of valuable information on virus genome co-evolution that has been mapped to internal branches of the tree.

In order to accomplish such an analysis, we need to extract the substitution map that has been generated by the QuickSelectionDetection batch file. The following is a code snippet that will write this substitution map to a file:

```plaintext
SetDialogPrompt ("Select a file to export data matrix");
fprintf (PROMPT_FOR_FILE, CLEAR_FILE, KEEP_OPEN);
```
for (i = 0; i < Abs(_bgm_data["MAP"]); i = i+1)
{
    fprintf (LAST_FILE_PATH, ",", (_bgm_data["MAP"])[i]+1); 
}
fprintf (LAST_FILE_PATH, "\n");
for (row = 0; row < Rows(bgm_data_matrix); row = row + 1)
{
    nodename = (_bac_tree_avl[row+1])["Name"];
    if (nodename[0][3] == "Node") { continue; }
    fprintf (LAST_FILE_PATH, nodename);
    for (col = 0; col < Columns(bgm_data_matrix); col = col+1)
    {
        fprintf (LAST_FILE_PATH, ",", bgm_data_matrix[row][col]);
    }
    fprintf (LAST_FILE_PATH, "\n");
}
fprintf (LAST_FILE_PATH, CLOSE_FILE);

The first column will contain the sequence names, which you can use to link each row of the substitution map to whatever agents of selection (or even phenotypes) that you have obtained for these sequences. When we were downloading HIV p24 sequences from the LANL website, we happened to include HLA genotypes into the sequence annotations. An example file containing a binary-state matrix corresponding to amino acid substitutions mapped to terminal branches leading to each sequence, as well as columns indicating the presence or absence of common HLA serotypes, is provided as a comma-delimited file named agents.csv. HLA serotypes are labeled in accordance with standard nomenclature, e.g., A24. (Note that codons in HIV p24 are numbered and prefixed with an ‘X’ in this example file, which was simply a consequence of merging the serotype and codon data in the statistical programming environment R, which does not permit variable names to begin with a number.)

In order to perform a BGM analysis outside of the QuickSelectionDetection batch file, we have provided a custom batch file called BgmAnalysis.bf. The options for this batch file are very similar to those raised by QuickSelectionDetection, with two important exceptions. First, you will need to specify a file containing a comma-delimited matrix where each column represents an integer-valued variable (substitution map at a given codon site, or presence/absence of an agent of selection, for example) and each row represents a terminal branch in the phylogeny. For each
column, the integer values must start at 0 and progress in increments of 1; in other words, a variable cannot skip 1 and go directly to 2. In the example matrix agents.csv, columns with HLA serotypes in the header contain a 0 to indicate that the serotype is absent and a 1 to indicate that it is present in the corresponding host. Second, the number of steps specified for a burn-in period is appended to the length of the chain, rather than indicating the number of steps in the chain to be discarded. For example, setting the chain length to 100,000 steps and the burn-in to 10,000 steps will now result in a total of 110,000 steps, of which the first 10,000 will be discarded before thinning. We recommend setting the maximum number of parents to 2, because this makes it easier to detect cases where the rate of evolution at one site is influenced by both an agent of selection (HLA serotype) as well as a second site in the genome.

Results from performing this analysis are displayed in Figure 1.6. Note that many epistatic interactions (edges between nodes representing codon sites) detected by our preliminary analysis (Figure 1.5) are recovered in this network as well; for example, 45/54/58. Edges between nodes representing HLA serotypes can generally be interpreted as alleles in linkage disequilibrium, i.e., MHC haplotypes, although we cannot rule out joint effects of the serotypes on a third variable that has not been incorporated into the analysis. We identify four edges between HLA serotypes and codon sites in HIV-1 p24. First, HLA serotypes B81 influence non-synonymous substitution rates at codon sites 45 and 54, which are within or adjacent to the known p24 epitope TPQDLNNTML (36). Second, codon site 110 is influenced by HLA serotypes A2 and B58, which is consistent with its membership in the known epitopes TSTLQEQIGW and STLQEQIGWM, respectively (29).

1.10 Exercises

Selection in the presence of recombination

In this exercise we will examine the CFVg.fas alignment and its GARD-partitioned counterpart CFVg-gard.nex for selection at individual sites, with and without correcting for recombination, using the fixed effects likelihood approach (FEL; (23)).

Lauch HyPhy, select Selection/Recombination from the standard analysis menu, then choose
QuickSelectionDetectionMF.bf. Use Universal genetic code, New Analysis, Custom nucleotide model, 012345 to specify the general time reversible model, 1 dataset to be analyzed, either CFVg.fas or CFVg-gard.nex for the input alignment, Estimate dN/dS only, the FEL method, 0.1 for the significance level for Likelihood Ratio Tests, All for branch option. Save results (a comma separated value) to a file (taking care keep partitioned and unpartitioned results in separate files).

As HyPhy performs the analysis, a typical output line may look like this:

Site 195 dN/dS = inf dN = 4.9848 dS = 0.0000 dS(=dN) 2.3353 Full Log(L) = -14.5463 LRT= 3.9208 p-value = 0.04769 *
P

Here, codon 195, has the maximum likelihood synonymous rate (dS) inferred at 0, and the non-syn rate (dN) - at 4.9848 (their ratio is infinite). The log-likelihood of the site with these parameters is -14.5463. The null model which forces dN=dS infers the value at 2.3353. The likelihood ratio test for non-neutral evolution has the test statistic of 3.9208 and the p-value of 0.04769 (which is significant at the specified level). The site is called positively selected (*P), because the test is significant and dN > dS. First, compare the list of sites reported as positively selected by the two analyses. Second, load the two resulting .csv files into a plotting program and draw a scatterplot of the p-values from the two analyses against each other. Do you think that there is an effect depending on whether or not we correct for the possible confounding caused by recombination?

Directional Selection

In this exercise we will use a Directional Evolution in Protein Sequences (DEPS, (27)) analysis to identify sites subject to directional positive selection in an alignment of 26 reverse transcriptase protein sequences (HIV_RT_AA_DEPS.nex) from HIV-1 infected patients sampled before and after antiretroviral therapy.

Lauch HyPhy, select Positive Selection from the standard analysis menu, then choose DirectionalREL.bf. Choose Reload (the initial model fitting take about 10-15 minutes, so the file you’ve downloaded contains the HIV_Between model (suitable for the analysis of HIV sequences) pre-fitted to this alignment), use the navigation box to find HIV_RT_AA_DEPS.nex, Unknown root. Hyphy will go to work and print some text to the console window. It will try every possible amino-acid residue and compute a p-value that some proportion of sites in the alignment are
evolving directionally towards that residue. Each of these models will be written to a file, e.g. `HIV_RT_AA_DEPS.nex.A`. For residues with significant p-values (after a multiple test correction), individual sites which may be evolving under directional selection are identified.

For instance, the text block below indicates that 1.8% of the sites show strong bias towards N ($p < 0.001$). Rates to N are 32.576 times faster than they are on the HIV_Between protein substitution model, leading to 53.828% frequency increase of residue N over the length of the tree.

[PHASE 12.1] Model biased for N

[PHASE 12.2] Finished with the model biased for N. Log-L = -2032.825

<table>
<thead>
<tr>
<th>Bias term</th>
<th>32.576</th>
</tr>
</thead>
<tbody>
<tr>
<td>proportion</td>
<td>0.018</td>
</tr>
<tr>
<td>Exp freq increase</td>
<td>53.828%</td>
</tr>
<tr>
<td>p-value</td>
<td>0.000</td>
</tr>
</tbody>
</table>

Three residues show evidence of directional evolution: W, N and V:

| Residues (and p-values) for which there is evidence of directional selection |
|-----------------------------|-----------------------------|
| W : 2.96368e-06             |
| N : 2.19281e-05             |
| V : 0.000763188             |

Site 103 is evolving directionally towards N, 184 towards V, and 210 and 219 towards W:

The list of sites which show evidence of directional selection (Bayes Factor > 20) together with the target residues and inferred substitution counts

Site 103 (max BF = 2.78192e+07)
Preferred residues: N
Substitution counts:
K->N: 8/N->K: 0
Site  184 (max BF = 13813.7)
Preferred residues: V
Substitution counts:
M->V: 10/V->M: 0
Site  210 (max BF = 1.96712e+07)
Preferred residues: W
Substitution counts:
F->L: 0/L->F: 1
L->W: 5/W->L: 0
Site  219 (max BF = 135.304)
Preferred residues: W
Substitution counts:
K->Q: 1/Q->K: 0
K->W: 1/W->K: 0

Co-evolution

Because we have gone through an example of using HyPhy to detect co-evolution in HIV-1 p24, let’s work through a different sort of example using the same data. (If you have just run the 1-parent Bayesian network analysis on these data using QuickSelectionDetection.bf, you will probably already have the likelihood function in memory under the name ‘lf’, and can skip the following.) Select the template batch file AnalyzeCodonData.bf from the Standard Analyses menu and choose the following options:

1. **Choose Genetic Code** — Select ‘Universal’.
2. **Tree Topology** — Select ‘p24_tree’, which will correspond to the tree that was included with the alignment in the data file.
4. Choose one of the standard models — Select MG94CUSTOM, which corresponds to the Muse-Gaut codon substitution model crossed with any nucleotide substitution model.

5. Model options — Select ‘Global’ to estimate one set of model parameters, such as the transition/transversion rate bias, for all branches in the tree.

6. Enter a PAUP*-style model specification string. For example, HKY85 is specified by the string 010010.

7. Use the tree included with the sequences in the p24.seq file by typing ‘y’ into the console window and hitting ENTER.

8. Branch Lengths — Select ‘Proportional to input tree’. Otherwise we will be estimating over 1000 branch length parameters in the tree, which is a very time-consuming analysis.

HyPhy will fit a codon substitution model to these data. This analysis will take at least a few minutes. When it is complete (HyPhy will spool a Newick tree string to the console), select the User Actions icon in the bottom-right corner of the console window (the icon is a pair of interlocked gears) and choose SimulateFromLF. You will be asked how many replicates to simulate. Type ‘1’ in the console window and hit enter. What we will be doing is simulating the evolution of codon sequences along the HIV-1 p24 phylogeny using the model parameters that we have just estimated from the data. Specify a file to save the simulated data to; it will be output in a NEXUS format, so you may want to use a ‘.nex’ file extension. Note that the filename that you specify is used as a prefix for all replicate simulations, which are distinguished from one another by an integer-valued suffix.

Now perform a BGM co-evolution analysis on the simulated alignment by executing the Quick-SelectionDetection.bf batch file and following the instructions in section 1.8 (use all the suggested default values). There should not be any sites identified as participating in an epistatic interaction, because the codon substitution model from which we simulated these sequences explicitly assumes that the evolution of each codon site is independent. Performing this analysis on simulated data is a useful negative control and assesses the false positive rate. When we performed this analysis,
we found that none of the edges in the network had a marginal posterior probability exceeding the 0.95 cutoff, and only two edges with probabilities greater than 0.9.

Now open the simulated alignment in a HyPhy data panel by selecting **Open Data File**... from the **File** drop-down menu. Select all sites by choosing **Select All** from the **Edit** drop-down menu and create a data partition object by choosing **Selection –> Partition** from the **Data** drop-down menu. This object will appear as a new row in the bottom-most field of the Data Panel. Set the **Partition Type** as ‘Codon’ and then translate the alignment into protein sequences by selecting **Aminoacid Translation** from the **Additional Info** sub-menu within the **Data** drop-down menu. Click on ‘All’ in the window that appears to translate all sequences in the alignment, and ‘Map to missing data’ in the following window to leave ambiguous nucleotides as unresolved.

A new data panel will appear with the protein sequence alignment. Select a small range of amino acid sites (about 30-40) by clicking and shift-clicking in the data panel and create a partition from this selection. Click on the magnifying-glass icon to open the **Data Operations** menu and select **Association [Fisher exact]**. Enter a significance level of 0.05 in the window that appears. (You may receive a warning that HyPhy will need to create X data partitions for significant clusters identified by this association test statistic — if so, hit ‘Cancel’.) This will mimic a pairwise correlation analysis of protein sequences that does not account for phylogenetic relationships nor confounding. Depending on which sites you select, you will observe some number of false positives; when we selected residues 40-80, we obtained 53 pairs with a *P*-value below 0.05, with some as low as $10^{-7}$. 
Bibliography


1.11 Tables

Table 1.1: HIV-1 reverse transcriptase drug resistance: episodic directional, episodic diversifying and constant diversifying selection. Note that the p-value for MEDS is obtained from a likelihood-ratio test (LRT) for episodic directional selection; FEEDS p-value is obtained from an LRT of the hypothesis $\beta^F > \alpha$ that tests for diversifying selection; and FEL is a test for constant diversifying selection run on Datamonkey. ‘-’ denotes a non-significant ($\alpha = 0.05$) p-value and ★ indicates no target residue because of lack of detection by MEDS.

<table>
<thead>
<tr>
<th>Site</th>
<th>Target</th>
<th>MEDS p-value</th>
<th>FEEDS p-value</th>
<th>FEL p-value</th>
<th>Resistance</th>
</tr>
</thead>
<tbody>
<tr>
<td>41</td>
<td>L</td>
<td>&lt; 0.0001</td>
<td>-</td>
<td>-</td>
<td>NRTI</td>
</tr>
<tr>
<td>74</td>
<td>★</td>
<td>-</td>
<td>0.001</td>
<td>0.007</td>
<td>NRTI</td>
</tr>
<tr>
<td>83</td>
<td>R</td>
<td>0.0004</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>103</td>
<td>N</td>
<td>&lt; 0.0001</td>
<td>0.007</td>
<td>-</td>
<td>NNRTI</td>
</tr>
<tr>
<td>184</td>
<td>V</td>
<td>&lt; 0.0001</td>
<td>0.02</td>
<td>-</td>
<td>NRTI</td>
</tr>
<tr>
<td>210</td>
<td>W</td>
<td>&lt; 0.0001</td>
<td>0.008</td>
<td>-</td>
<td>NRTI</td>
</tr>
<tr>
<td>215</td>
<td>Y</td>
<td>&lt; 0.0001</td>
<td>&lt; 0.0001</td>
<td>&lt; 0.0001</td>
<td>NRTI</td>
</tr>
<tr>
<td>219</td>
<td>W</td>
<td>0.0017</td>
<td>-</td>
<td>-</td>
<td>NRTI accessory</td>
</tr>
</tbody>
</table>
Figure 1.1: (A). Phylogenetic incongruence caused by the presence of a recombinant sequence in an alignment. Sequence R is a product of homologous recombination between sequences A and B. Phylogenies reconstructed from sequences A,B,R and an outgroup sequence (O) will differ based on which part of the alignment is being considered – to the left of the breakpoint, R clusters with A, whereas to the right of the breakpoint R clusters with B. (B). GARD analysis of the Cache Valley Fever Virus glycoprotein.
Figure 1.2: The effect of recombination on inferring diversifying selection. Reconstructed evolutionary history of codon 516 of the Cache Valley Fever virus glycoprotein alignment is shown according to GARD inferred segment phylogeny (left) or a single phylogeny inferred from the entire alignment (right). Ignoring the confounding effect of recombination causes the number of nonsynonymous substitutions to be overestimated. A fixed effects likelihood (FEL, (23)) analysis infers codon 516 to be under diversifying selection when recombination is ignored ($p = 0.02$), but not when it is corrected for using a partitioning approach ($p = 0.28$).
Figure 1.3: A phylogeny of reverse transcriptase sequences. Foreground branches which lead to post-treatment sequences are colored red.

Figure 1.4: *HyPhy* BGM diagnostics. (*left*) A graph depicting a thinned Markov chain Monte Carlo sample from the posterior probability distribution of Bayesian networks given the HIV-1 p24 example data. (Caveat: posterior values are labeled as ‘LogL’, which is an abbreviation of log-transformed likelihood.). (*right*) A histogram summarizing the edge marginal posterior probabilities from the same analysis.
Figure 1.5: A graph depicting compensatory interactions inferred from the alignment of HIV-1 subtype C gag p24 sequences. Each square node represents a position in the gp41 protein sequence that participates in at least one interaction. The arrows (edges) representing those interactions are annotated with the fraction of graphs sampled in chain sample that contain the edge.

Figure 1.6: A Bayesian network inferred from the joint distributions of codon site-specific substitutions mapped to terminal branches of the HIV-1 p24 phylogeny (open nodes), and HLA serotypes presented by the corresponding host environments (filled nodes). A marginal edge posterior cutoff of 0.9 was used to generate this consensus network. Edges between HLA serotypes and HIV p24 codon sites are highlighted in bold.